Amendments to the Specification:

Please replace the paragraph beginning at page 1, line 2 as with the following amended paragraph:

This application is related to U.S. Application Serial No. [[_____]] 10/625,897 titled "MEMORY MANAGEMENT FRAME HANDLER" which is being filed concurrently with this application.

Please replace the paragraph beginning at page 3, line 25 as with the following amended paragraph:

FIGS. 4A-4D [[is a]] are diagrams of InfoType tree structures indexing attributes within a fast query system.

Please replace the paragraph beginning at page 3, line 27 as with the following amended paragraph:

FIGS. 5A-5D [[is a]] are diagrams of InfoCourse tree structures indexing data records within a fast query system.

Please replace the paragraph beginning at page 5, line 16 as with the following amended paragraph:

Freeform text search systems are sometimes implemented by creating structured data representing a freeform record. Then, structured data techniques may be applied to the freeform records. For example, a list of words in a web page may be used to create structured data linking the words to the web page. The structured data may be indexed and stored such that a user may perform queries to identify web pages containing a particular word.

Please replace the paragraph beginning at page 6, line 1 as with the following amended paragraph:

Referring to FIG. 1, a fast query system 100 includes a data query module 102, a data maintenance module 104, and a data store 106. The data query module 102 receives and processes queries 108. Data queries may be formulated in a proprietary query language, or in a standard query language such as the structured query language (SQL). The data maintenance module 104 may perform maintenance tasks such as data loads, backups, indexing, and optimization. The data maintenance module 104 may be used to load data from, for example, a flat file or RDBMS into the fast query system 100. Loaded data is stored in the data store 106 in a format to facilitate fast responses to queries as is described below with respect to FIGS. 2, 3, and 4A-4D.

Please replace the paragraph beginning at page 7, line 8 as with the following amended paragraph:

Referring to FIGS. 4A-4D, InfoTypes 310 and InfoCourses 308 may be implemented using any indexing structure including tree structures or hash tables. For example, conventional balanced tree structures such as red-black trees and AVL trees may be used. FIGS. 4A-4D ~~illustrates~~ illustrate InfoTypes 310 using balanced binary trees used in an AVL index. In this example, each node within the InfoType 310 includes a left and right pointer. A node's left pointer identifies elements less than the node and a node's right pointer identifies elements greater than the node. For example, "Nash" is less than "Smith" and "Zimmerman" is greater than "Smith." The numbers on each node in FIGS. 4A-4D represent the node's InfoType 310. The number "1" represents the last name attribute 212, "2" represents the first name attribute 214, "3" represents the address attribute 216, and "4" represents the city attribute 218. Each attribute is represented by a corresponding InfoType 310.

Please replace the paragraph beginning at page 7, line 19 as with the following amended paragraph:

Referring to FIGS. 5A-5D, InfoCourses 308 represent data records 202 in a data query system 100. The fields 210 within a data record 202 may be indexed to facilitate retrieval of a specific field 210. For example, FIGS. 5A-5D [[shows]] show four AVL trees indexing fields

210 in an InfoCourse 308. Each field 210 is represented by an InfoCell 312. Alternatively, any other indexing technique may be used including, but not limited to, red-black trees, B-trees, or hash tables. In this example, an arbitrary number corresponding to an InfoType 310 is used to index the data record 202 within an InfoCourse 308. In ~~the first tree in~~ FIG. 5A, the root node, numbered "3," stores the address attribute 216, "123 Main St." To its right is the city attribute 218, "Smallville," because its corresponding number "4" is greater than "3." To its left is the first name attribute 214, "John," with a corresponding number "2" less than "3." Finally, the last name attribute 212, "Smith," with a corresponding number "1" is to the left of the first name attribute 214.

Please replace the paragraph beginning at page 8, line 6 as with the following amended paragraph:

FIG. 7 shows an exemplary InfoCell 312 data structure that may be used by InfoCourses 308 and InfoTypes 310. The InfoCell 312 includes a left InfoType pointer 702 and a right InfoType pointer 704. These pointers are used to define an InfoType 310 tree structure. For example, in FIG. 4A the InfoCell 312 for the last name attribute 212 "Smith" includes a left InfoType pointer 702 to the "Nash" InfoCell 312 and a right InfoType pointer 704 to the "Zimmerman" InfoCell 312. Some InfoCell 312 data structures do not need a left InfoType pointer 702 or a right InfoType pointer 704. Blank InfoType pointers 702 or 704 may point to the null value or may reference an anchor node of the InfoType 310. An anchor node points to the root node of the InfoType 310 so the system may begin a tree traversal or so that the system may identify when the bottom of the tree has been reached. Additionally, the root node of the tree may be located by traversing the tree until the anchor node is reached. Similarly, a left InfoCourse pointer 712 and a right InfoCourse pointer 714 are used to define an InfoCourse 308 tree structure. Each InfoCourse 308 also may include an anchor node.

Please replace the paragraph beginning at page 11, line 10 as with the following amended paragraph:

Referring to FIG. 11, each frame 902 may be further divided to store instances (1102, 1104, 1106, and 1108) of a particular data type. In this example, an InfoType 310, from the example discussed with reference to FIGS. 4A-4D, is an AVL tree storing an attribute from a group of data records. Each node in the AVL tree is an instance of an InfoCell 312 (as shown in FIG. 7). These InfoCells 312 are stored in portions of a frame 902 within a BigPage 804.

Please replace the paragraph beginning at page 11, line 15 as with the following amended paragraph:

In this example, an InfoCell 312 representing an attribute of a data record having the value "~~Meuller~~ Mueller" is stored in instance 1102 of frame 4, an attribute having the value "Nash" is stored in instance 1104, an attribute having the value "Smith" is stored in instance 1106, and an attribute having the value "Zimmerman" is stored in instance 1108. Because each of these instances is stored within a BigPage 804, it is unnecessary to consume overhead 1004 for each of the objects. Additional data may be stored in the remaining available space 1110.

Amendments to the Drawings:

The attached replacement sheets of drawings include changes to Figs. 4-5 and replace the original sheets including Figs. 4-5.

Figure 4 is amended to distinguish four InfoType tree structures (FIGS. 4A-4D) indexing attributes within a fast query system. Figures 4A and 4B are found on one replacement sheet. Figures 4C and 4D are found on a separate replacement sheet.

Figure 5 is amended to distinguish four InfoCourse tree structures (FIGS. 5A-5D) indexing data records within a fast query system. Figures 5A and 5B are found on one replacement sheet. Figures 5C and 5D are found on a separate replacement sheet.

Attachments following last page of this Amendment:

Replacement Sheets (4 pages)
Annotated Sheet Showing Change(s) (2 pages)